|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre de la práctica** | Reporte de Apuntadores en C | | | **No.** | **15** |
| **Asignatura:** | **Métodos Numericos** | **Carrera**  **:** | **Ingeniería en**  **Sistemas**  **Computacionales** | **Duración de la práctica (Hrs)** | **2hra** |

**Nombre del alumno:** Carmen Pérez Soto

**Grupo:342**

**I. Competencia(s) específica(s):**

**I. II. Lugar de realización de la práctica (laboratorio, taller, aula u otro): En casa**

1. **Material empleado:**

**\* lamina de ejrcicios**

**\*VSCodium**

**\*word**

1. **Desarrollo de la práctica:**

Apuntadores

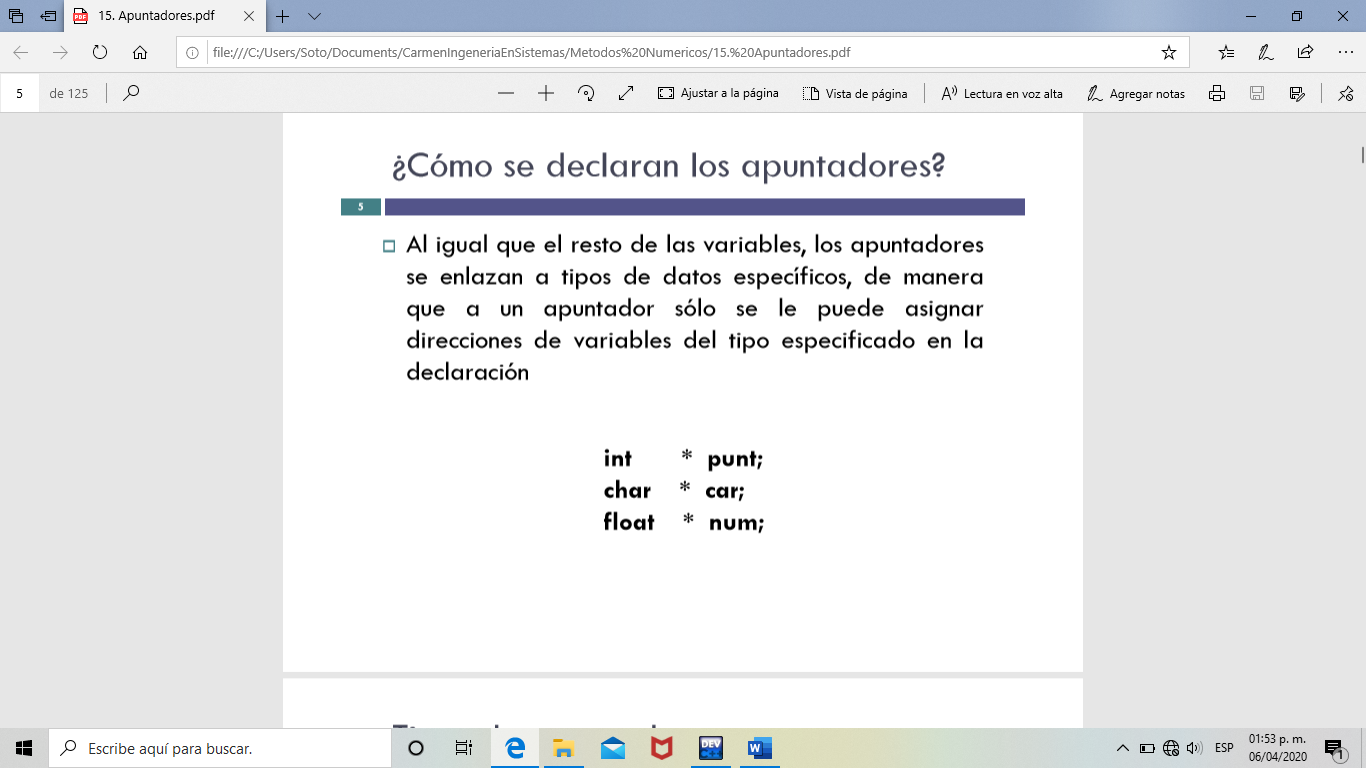
Antes de empezar tenemos que saber que es un apuntador y tambien a su vez este para que nos sirve esta practica ya se realizo desde Linux en Porrot OS.

**¿Qué es un apuntador?**

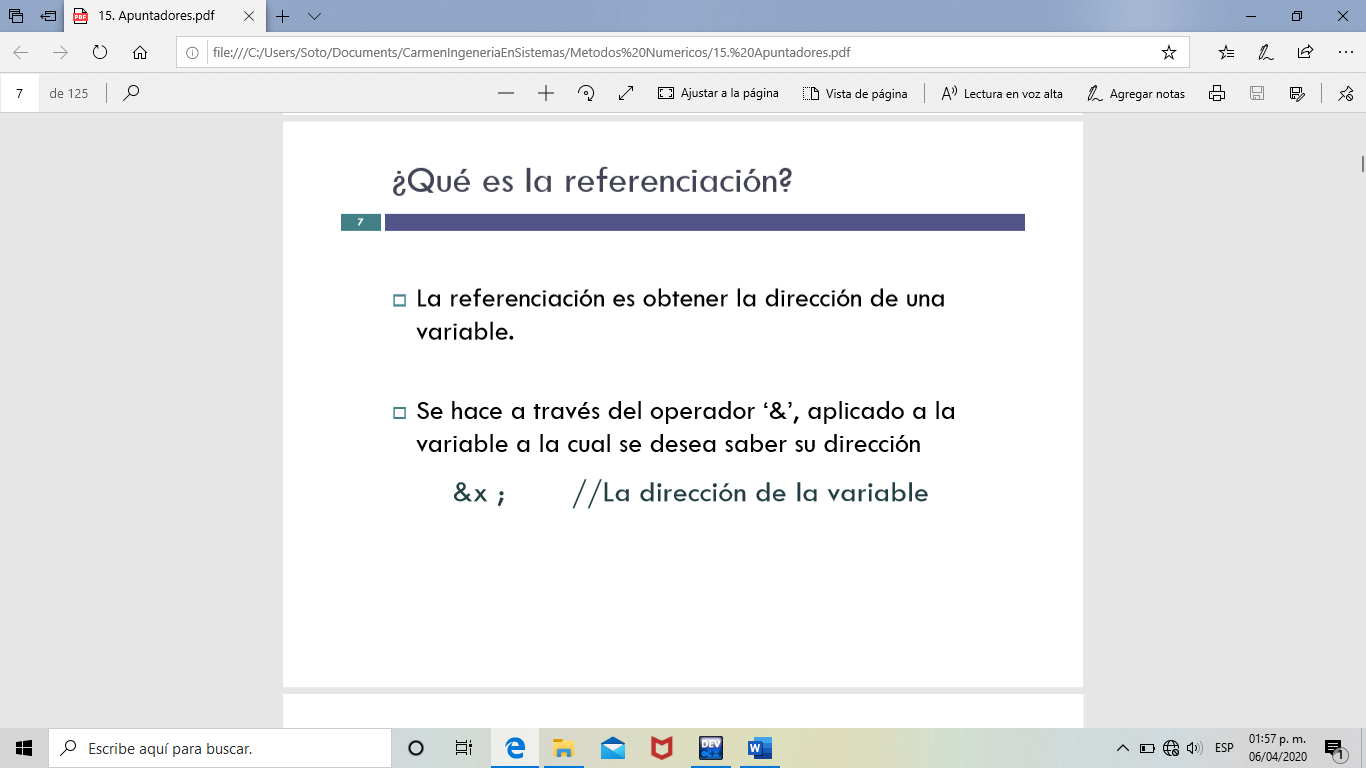
Un puntero es un objeto que apunta a otro objeto. Es decir, una variable cuyo valor la dirección de memoria de otra variable.

En C no se debe indicar numéricamente la dirección de la memoria, sino que se usa una etiqueta que conocemos como variable.

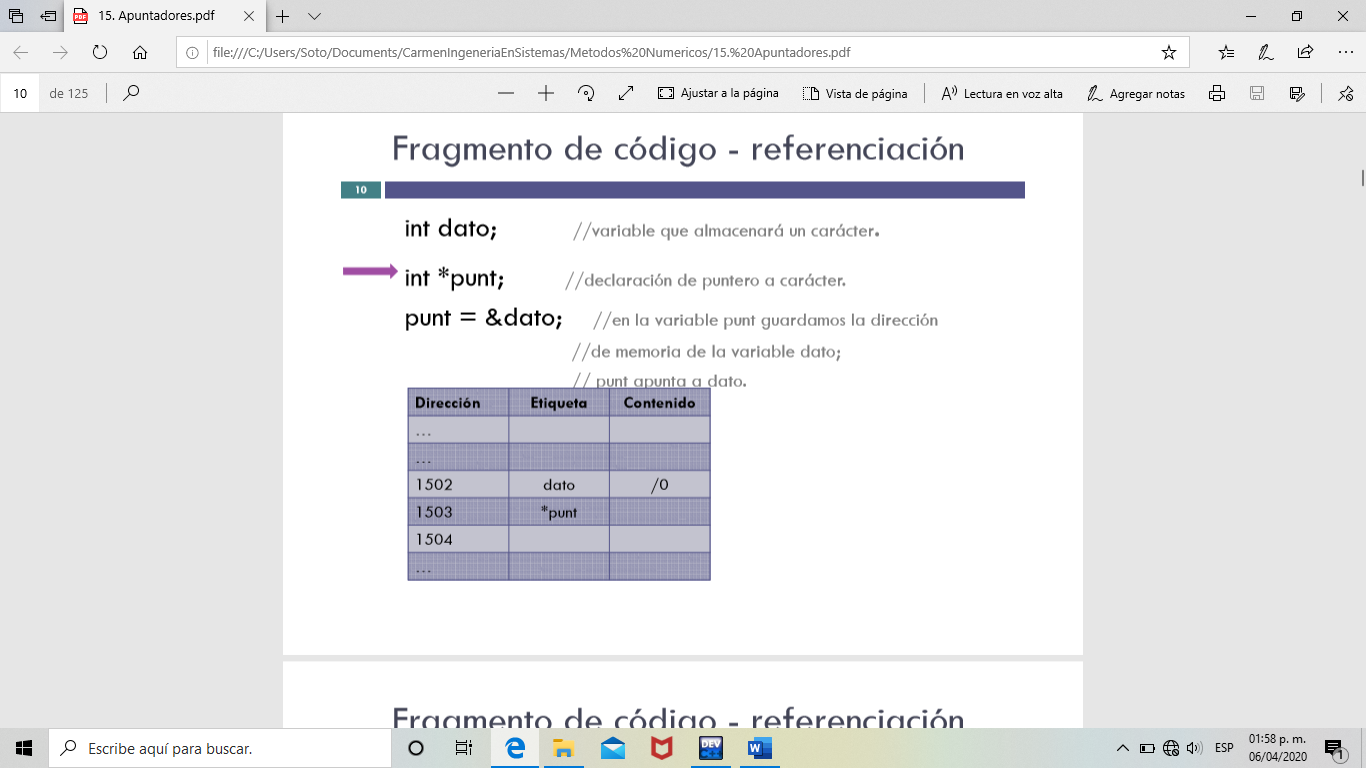
Otra cosa que debemos saber es como declarar un apuntador en la parte de abajo se muestra cómo se declara:



¿Qué es la referenciación?

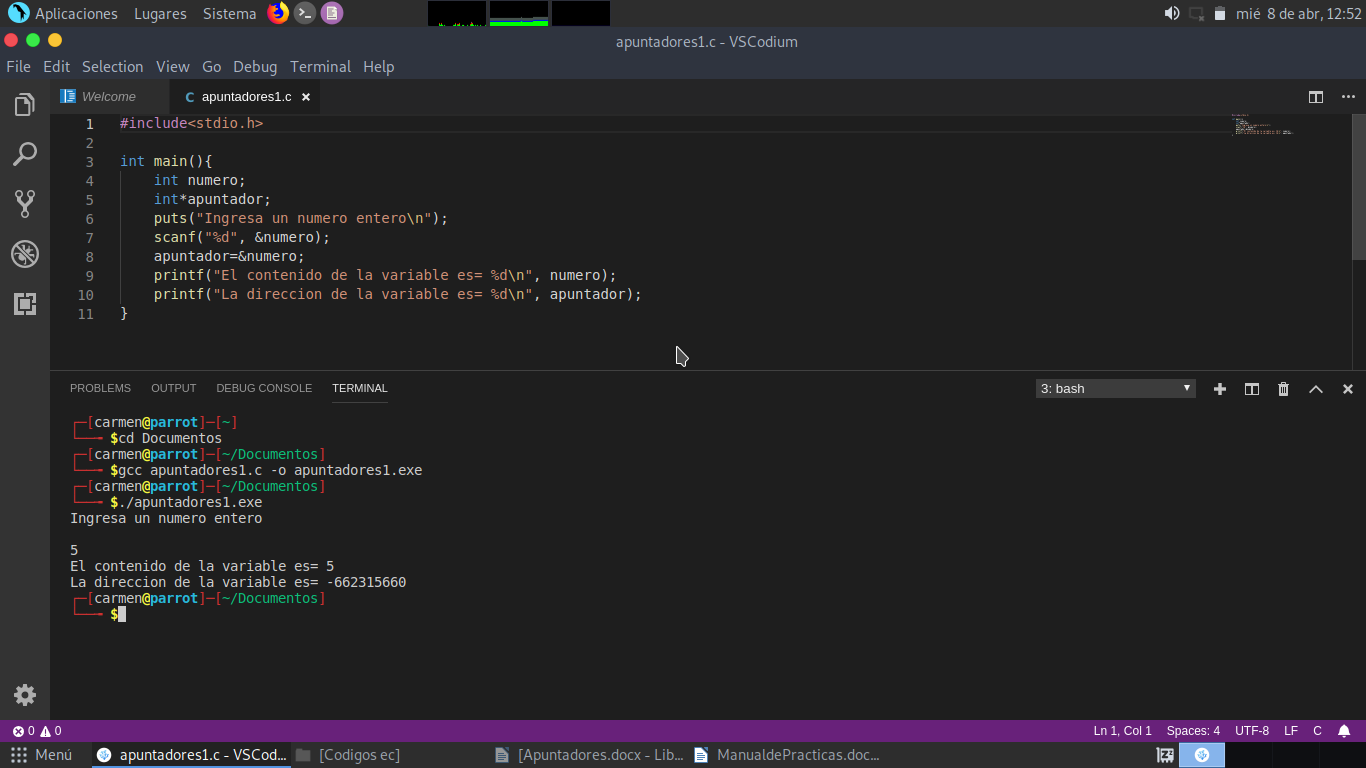


Ejercicio 1

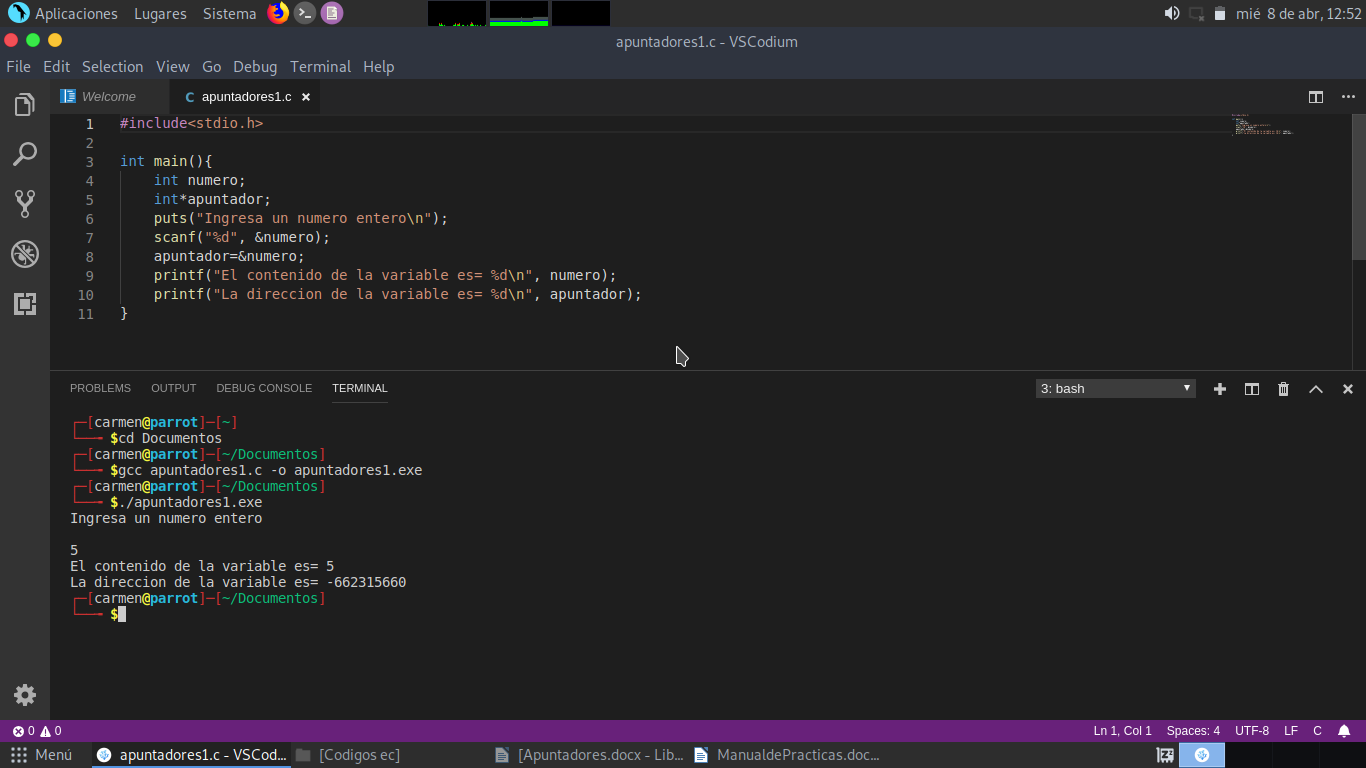


Bueno para esto hice un programa primero declare una variable llamada numero de tipo entero, despues declare un apuntador de tipo entero que es el tipo de dato, asterisco y el nombre que en este caso le puse apuntador.

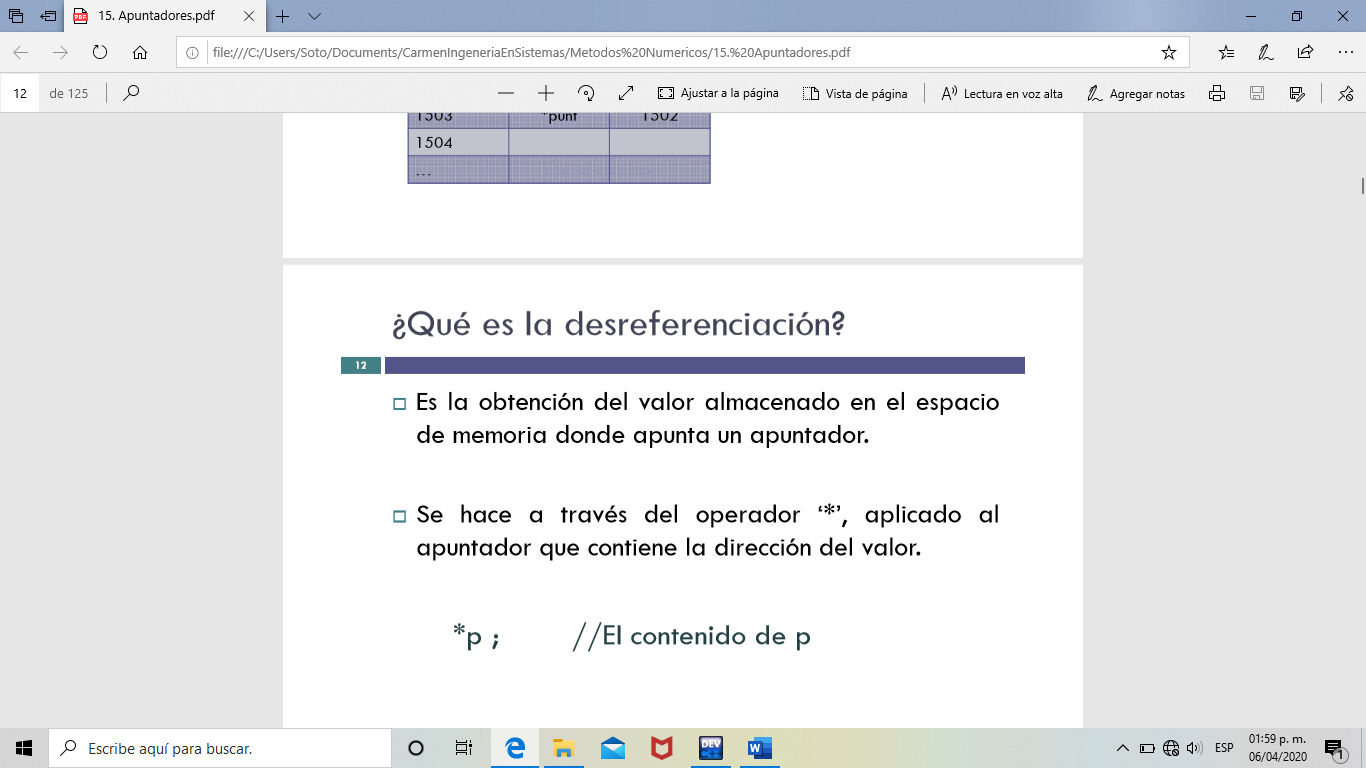
Despues con ayuda de la función puts mande un mensaje pidiendole al usuario que ingresara un numero entero, posteriormente con la funcion scanf pedi que el usuario desde teclado ingresara el numero. A continuación en la variable apuntador guarde la direccion de la variable numero para finalizar mande dos mensajes uno para dar a conocer su contenido y en otra para saber su dirección.



Despues guardo y ejecuto mi programa, como se muestra abajo se muestra que pide que engrese un numero entero y se ingresa 5, posteriormente por medio de mensajes se muestra lo que contiene la variable numero y su dirección.

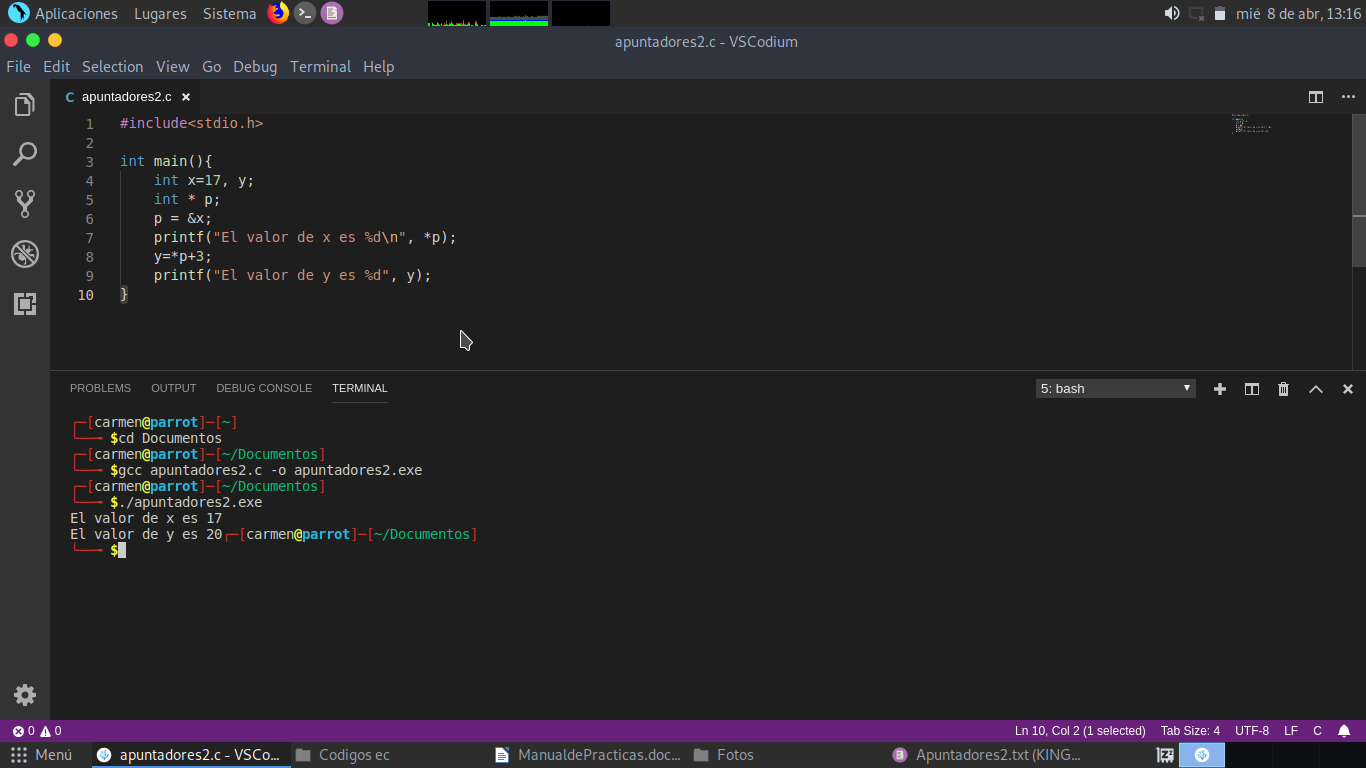


Ejercicio 2

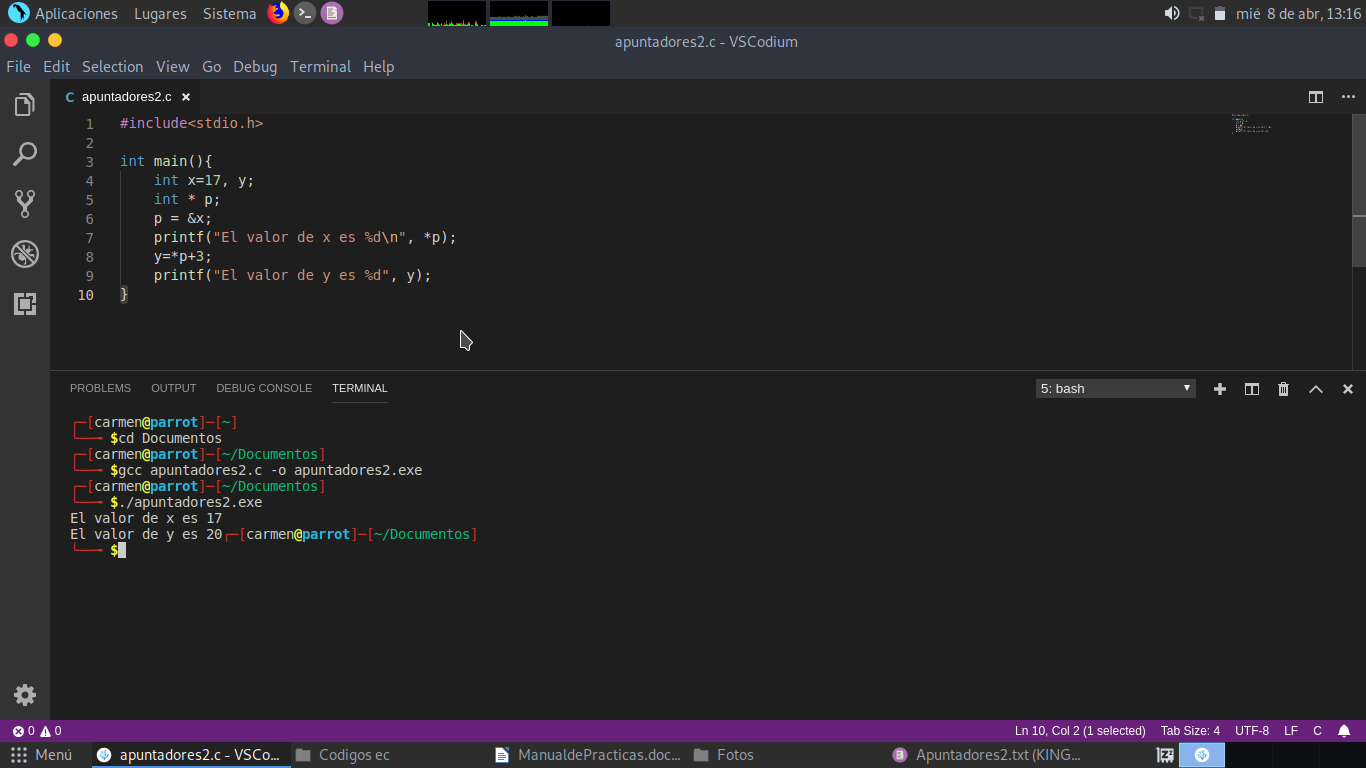


Bueno aquí en este ejercicio coloque dos variables de tipo entero una la inicialice a x con un valor de 17. Declare un apuntador llamado p, porteriormente a la varaible p le guarde la direccion de x, mande un mensaje con el valor de x con ayuda de este signo \* y de p que es el apuntador muestra el valor de x osea 17, posteriormente y va a guardar lo que tenga \*p y le sumare 3 posteriormente se imprime lo que contiene y.

Lo que debe mostrar a la hora de correrlo x = 17 , y=20 guardamos y verificamos si si nos da eso.

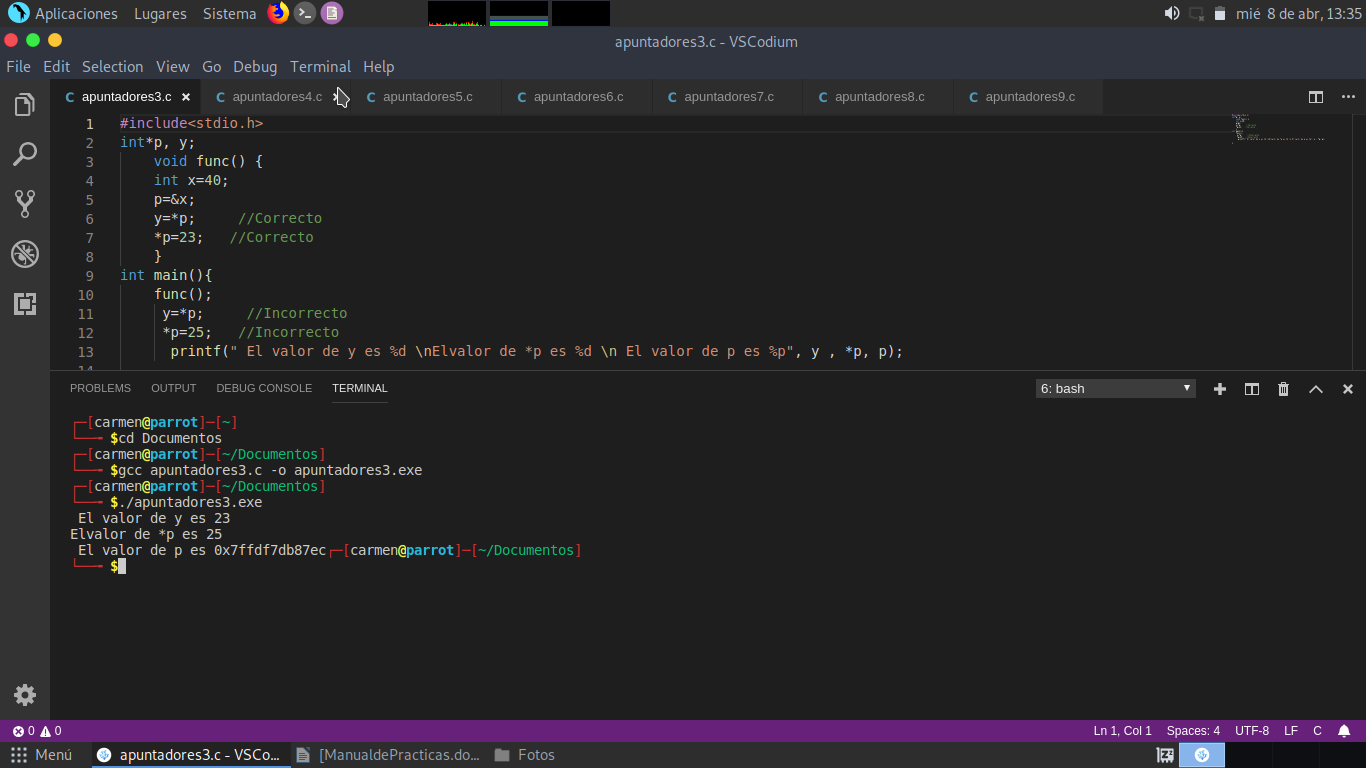


Como se muestra en la imagen de abajo si nos da los resultados esperados, este fue un ejercicio que muestra como se utiliza la desreferenciacion.

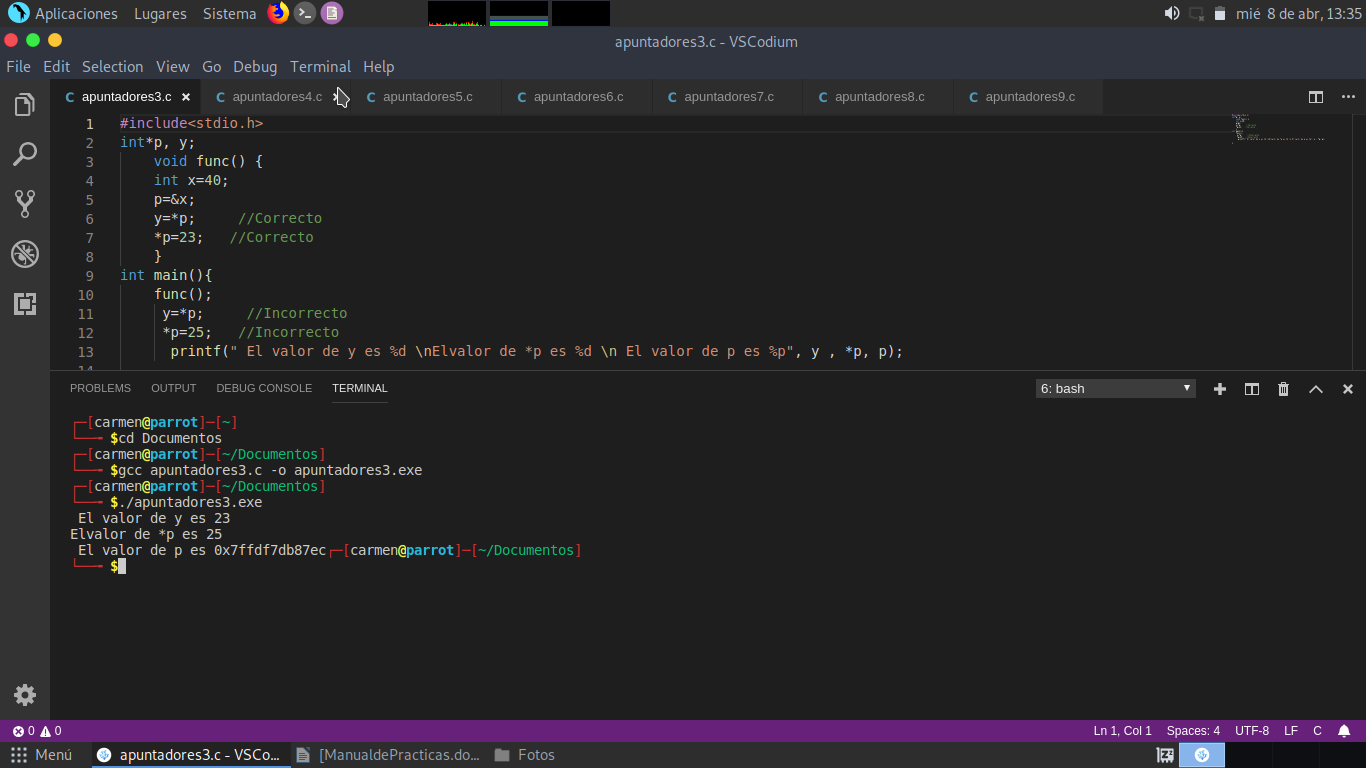


Ejercicio 3

En este ejercicio se mostarar como hacemos uso de las funciones y vamos a dar los valores de y de \*p y de p. Para esto fuera del emtodo main colocamos dos variables de tipo entero una es el apuntador y otra es y, dentro del metodo main colocamos funnc(); indica que primero se va a ir a la funcion nos vamos a la funcion dentro de esta esta un variable de tipo entero llamanda x e inicializada en 40, despues p va a guardar la direccion de x, y va hacer igual al valor del apuntador que en este caso gauda 40 posteriormente \*p va hacer igual a 23, ciuando termine sale y se va de nuevo al metodo principal y dice que y es igual a \*P que este vale 23, \*p es iagul a 25, por ultimo mando a imprimir los valores.



En la ventana de abajo se ve cuando lo ejecuto y me da los resultados de las variables, podemos ver que los resultados son correctos.se va el valor de y, \*p y el valor de p.

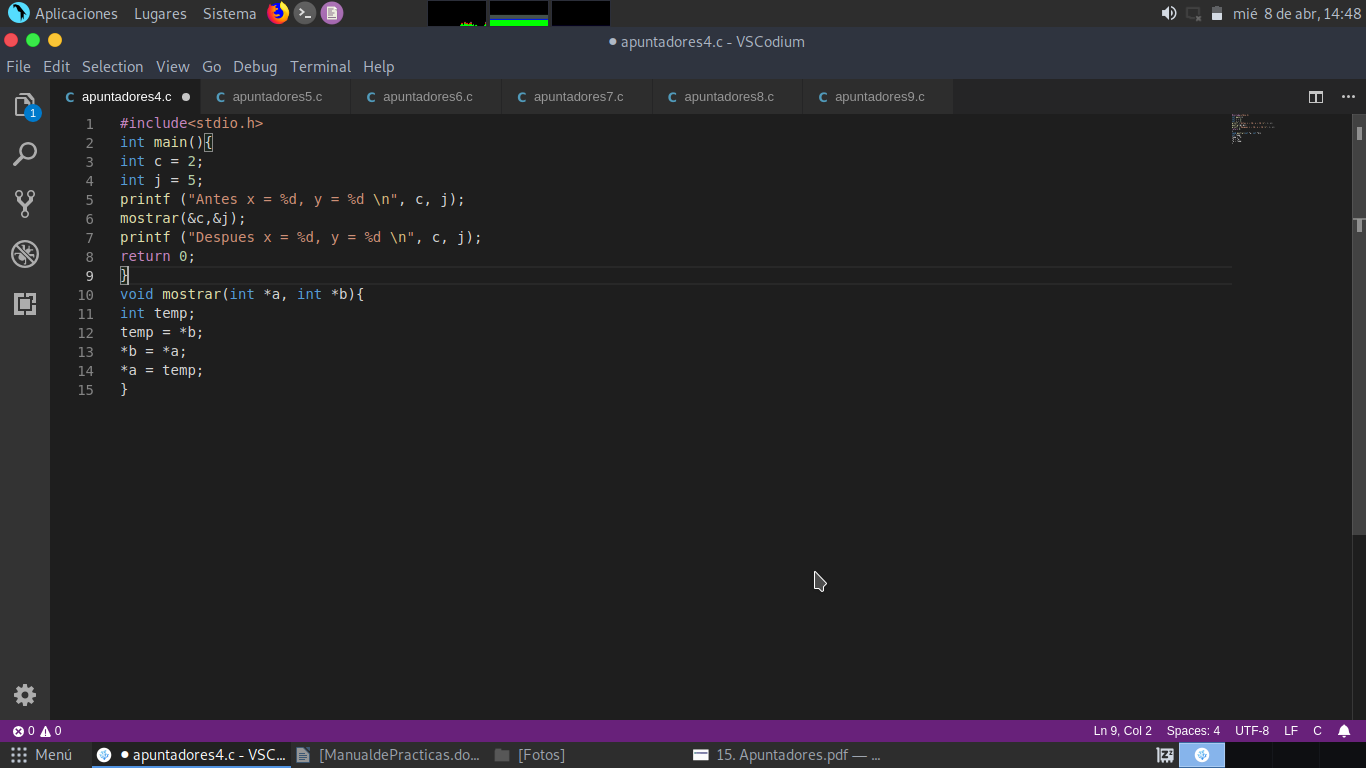


Ejercicio 4

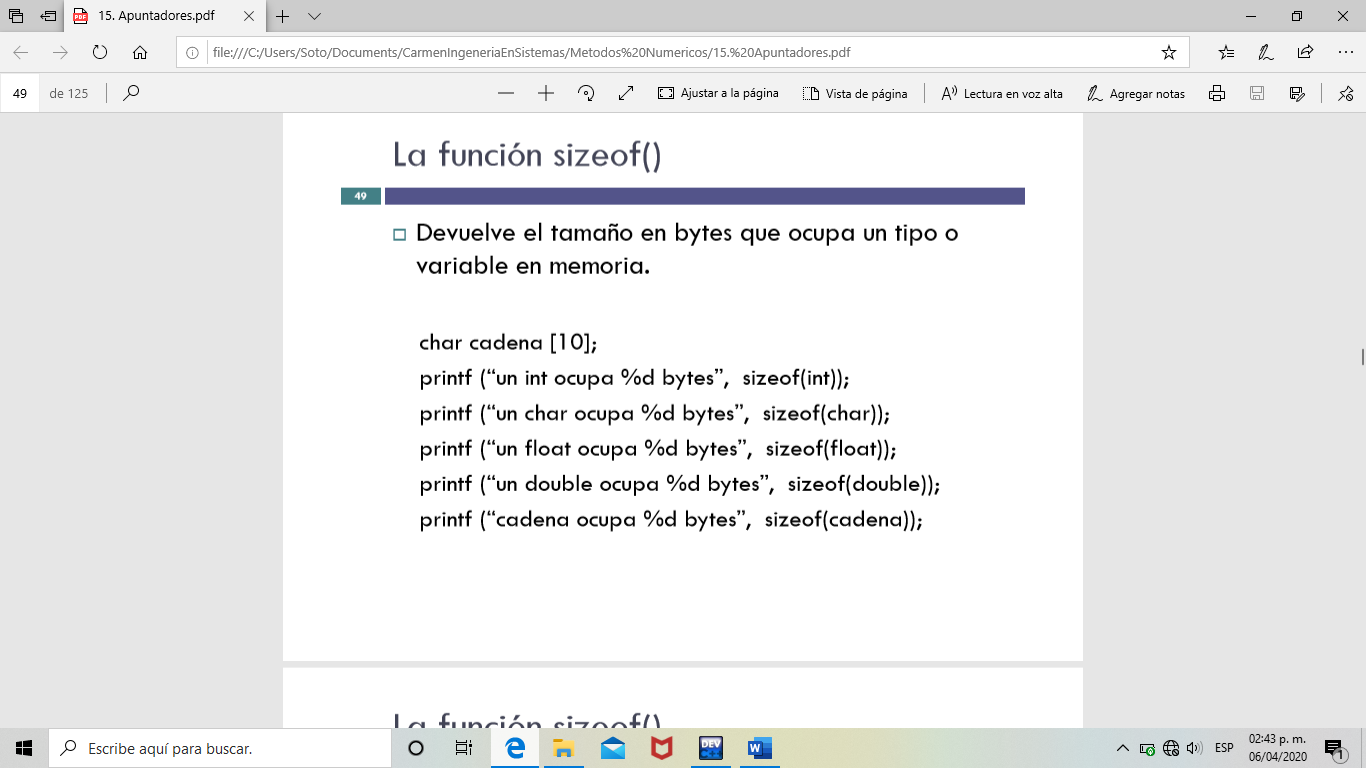
En este ejercicio vamos a cambiar los valores de c y j se mostrara el valor de cy j e invertidos despues, para hacer esto vamos hacer uso de una función y de apuntadores.

Primero declaramos las dos variables de tipo entero les damos vamores ya que c vale 2 y j vale 5, vamos a mandar a imprimir lo que contiene cada variable, posteriormente se muestra el nombre de la funcion con sus parametros, para esto nos vamos a la funcion, vemos que dentro de ella tiene dos apuntadores de tipo entero uno llamado a y otro b, declare una variable de tipo entero para rempalzar el valor de la variable.

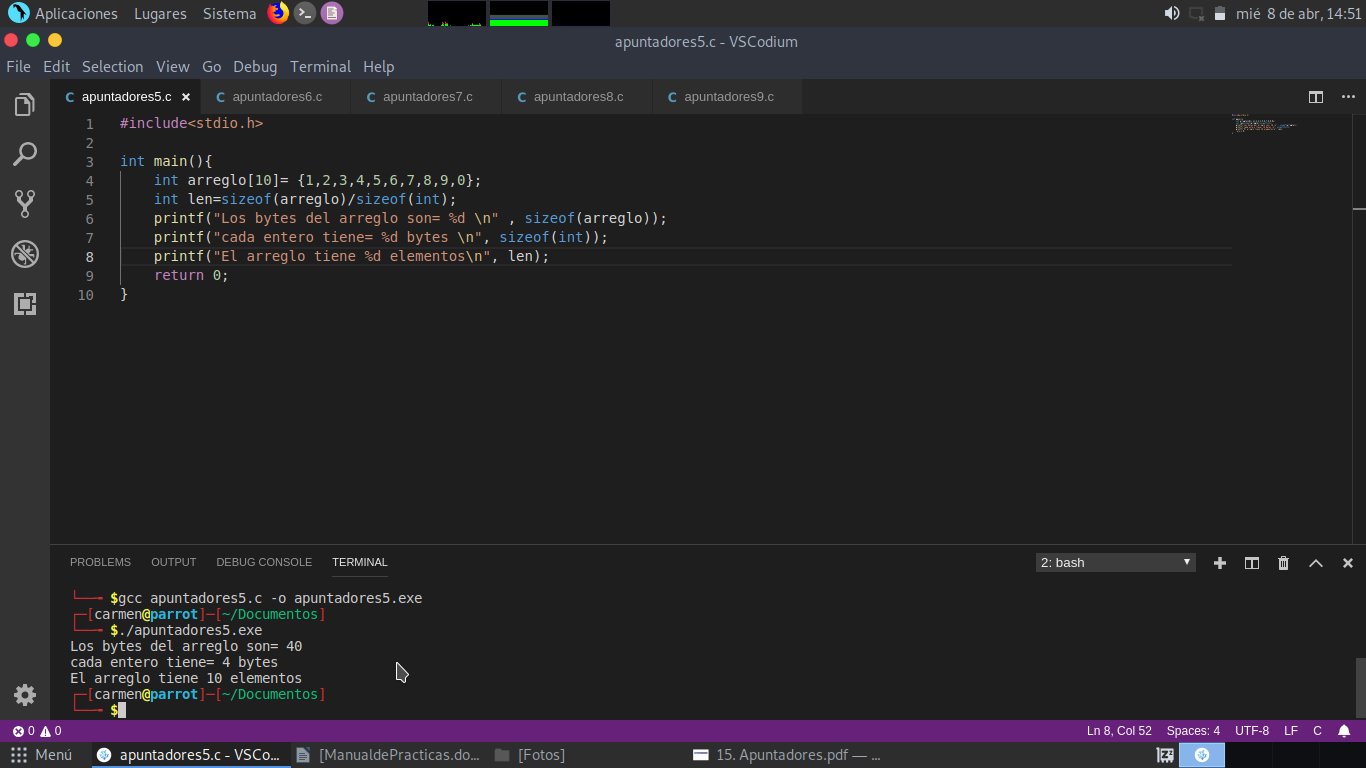
Temp va a guardar el valor que tiene \*b que es 5, despues \*b va hacer igual a \*a que es 2, despues \*a es igual a lo que tiene temp termina de ejecutarse eso y regresa al metodo main ya despues va a imprimir el valor de c y j los valores de c=5 y j=2 ahorita que se mande a imprimir se vera si el proceso se hiso correctamente.



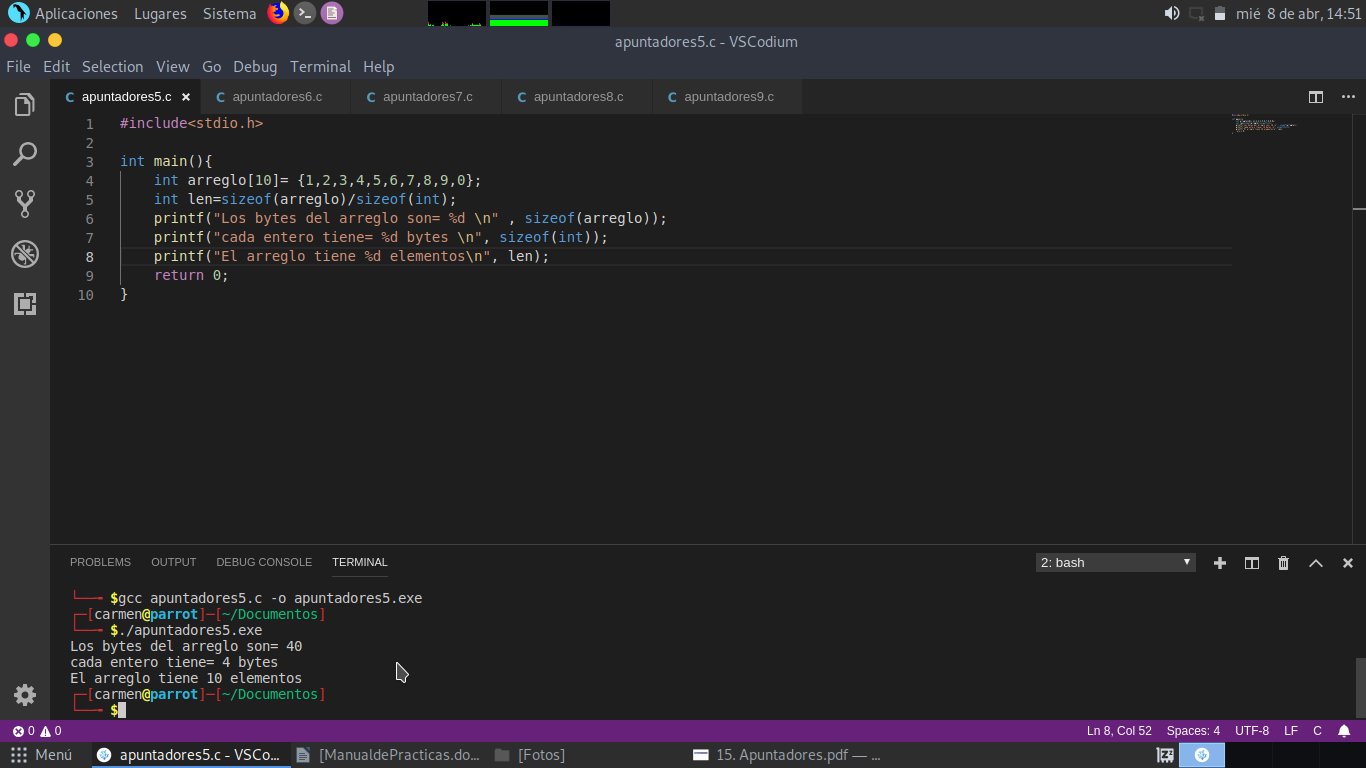
Ejercicio 5

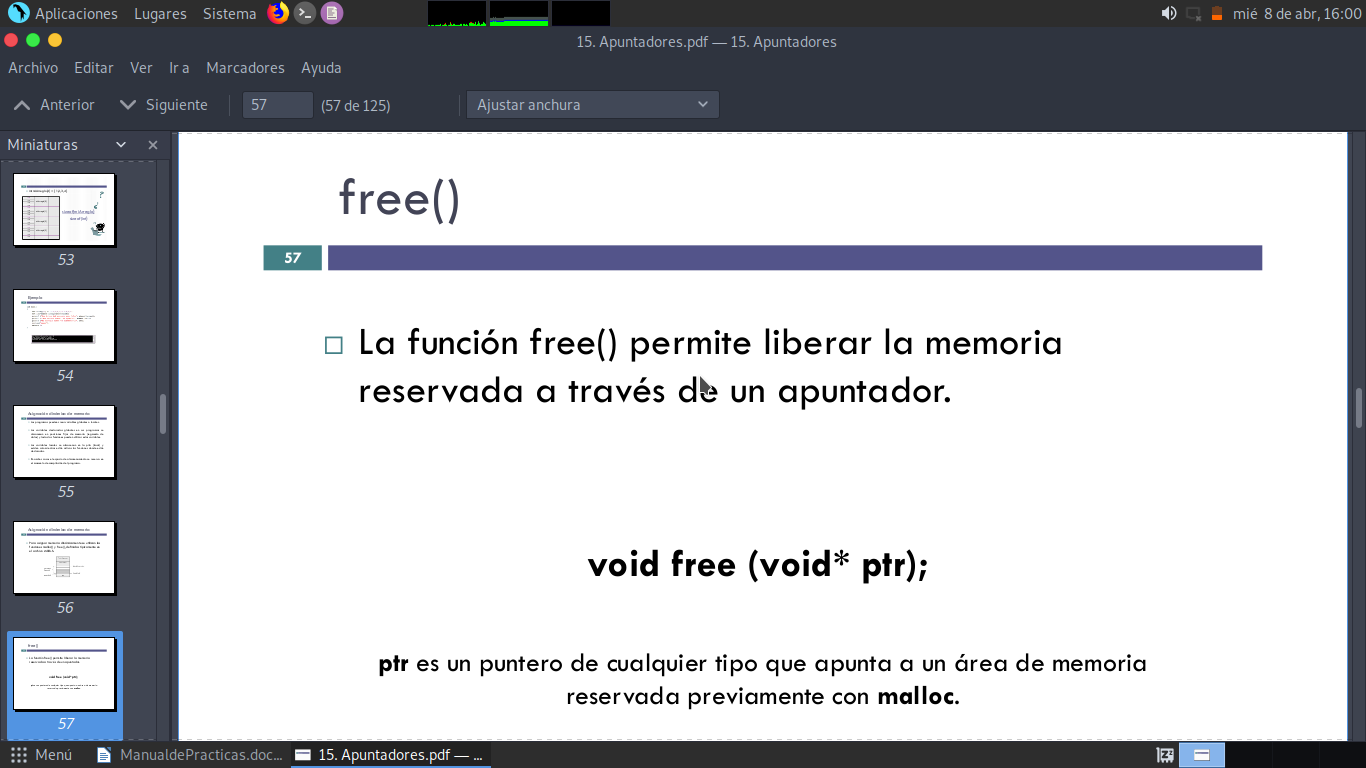


para esto se realizo el siguiente ejercicio para ver como es que funciona, declaramos un arreglo con numeros ya definidos que en este caso son 10 de tipo entero, debajo esta una variable que nos devolvera el tamaño en bytes, que tamaño tiene el arreglo y el numero de elementos que tiene.



En la imagen de abajo se muestra los bytes que ocupa el arreglo, cuantos bytes tiene cada entero y cuantos elementos tiene este.



Ejercicio 6

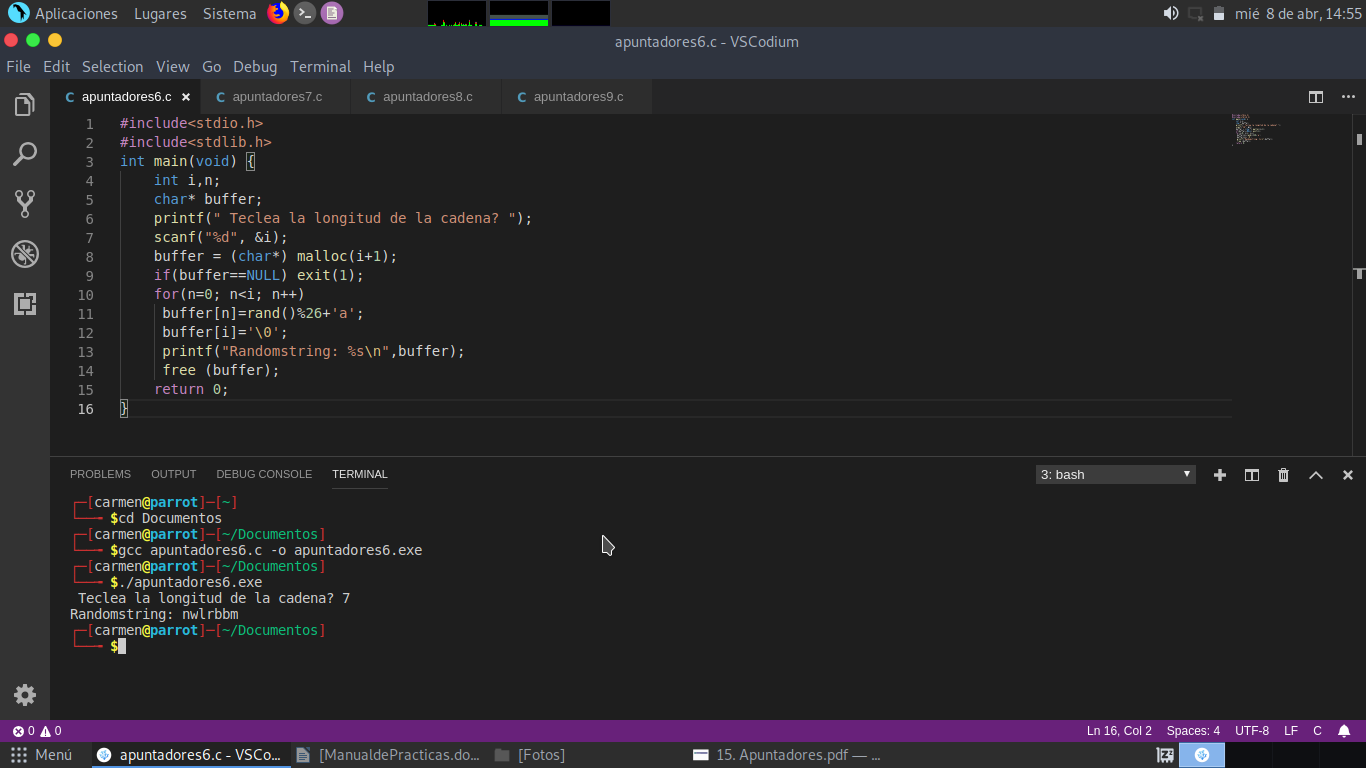
malloc()

La función malloc() reserva memoria y retorna su dirección, o retorna NULL en caso de no haber

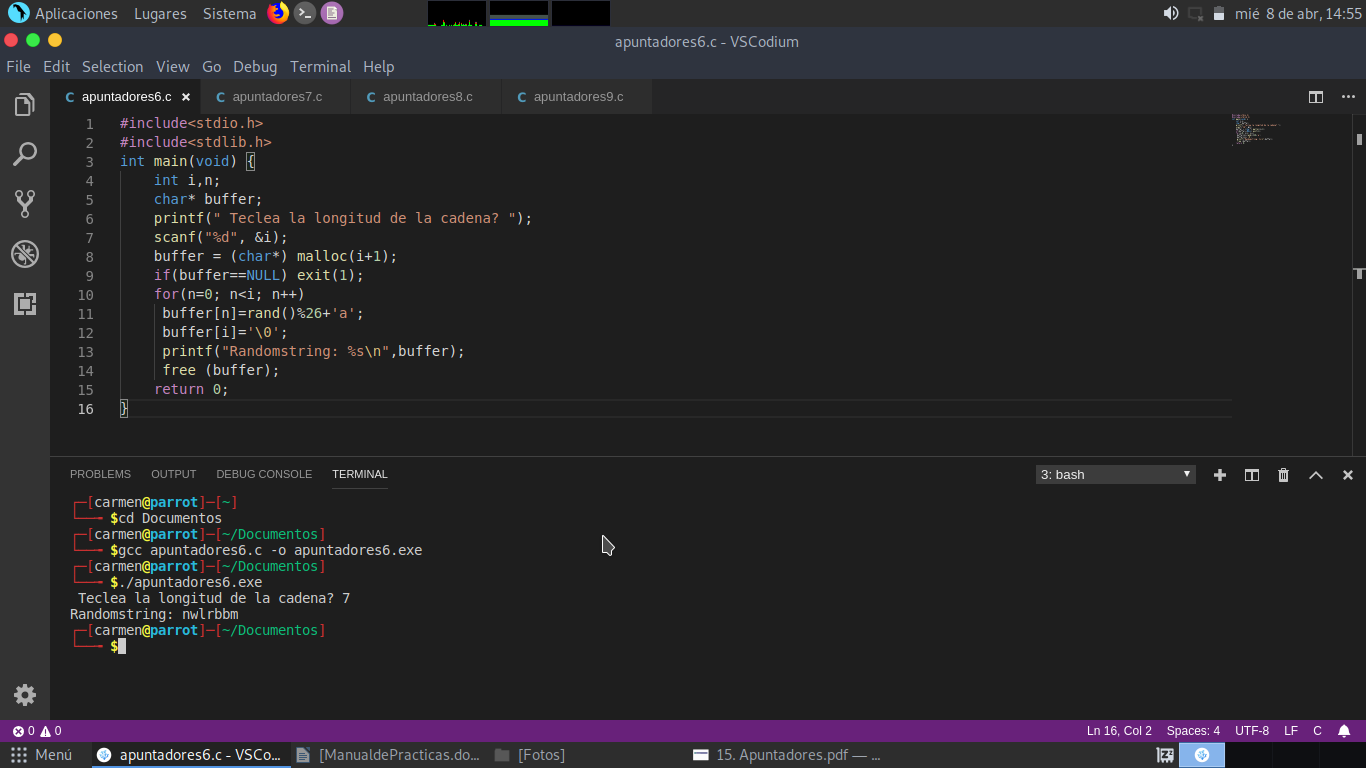
conseguido suficiente memoria.

Void \*malloc(size\_t tam\_bloque)

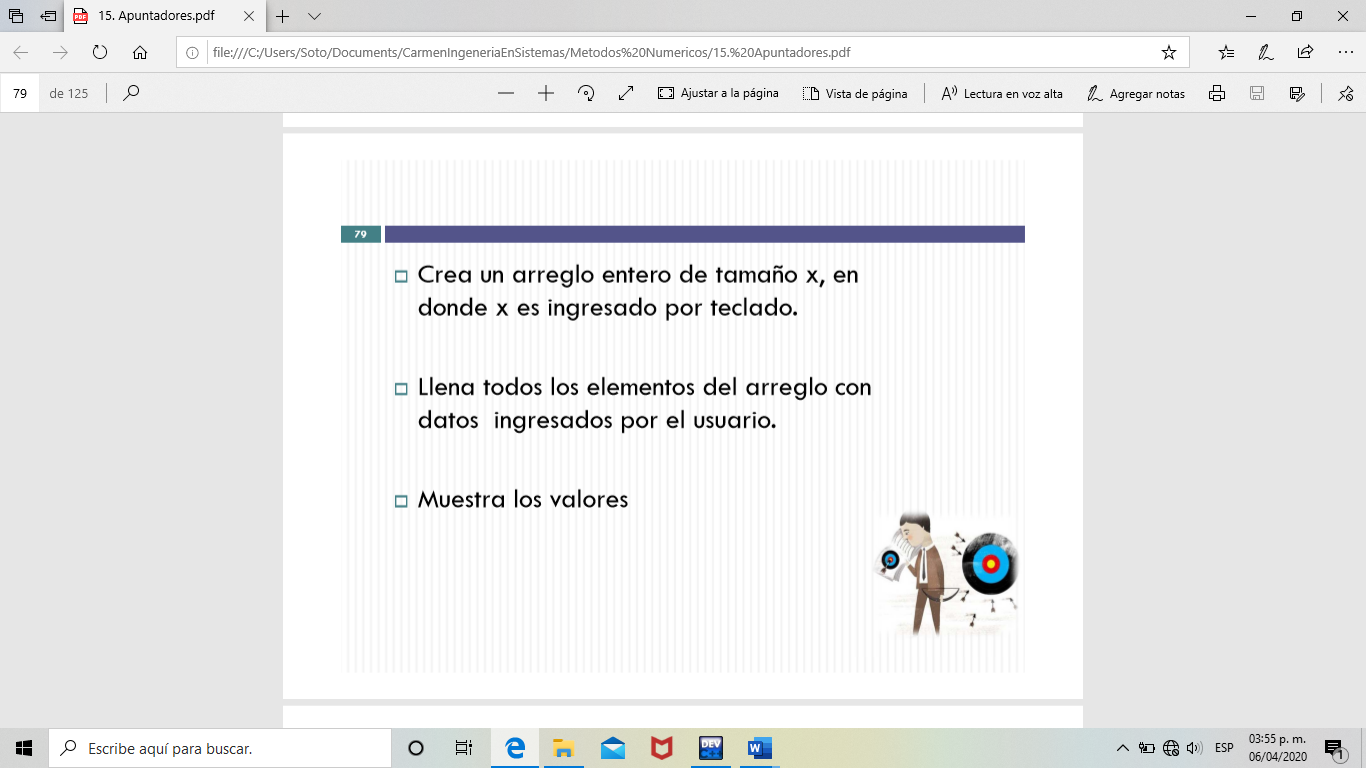
para ver aplicado estos se realizo el siguiente ejercicio se declaran dos variables de tipo entero despues un apuntador de tipo char llamado buffer, con ayuda del printf mandamos un mensaje pidiendole al usuario que ingrese la longitud de la cadena por medio del teclado con ayuda del scanf y pues ya abajo se usa lo que esta arriba me cuenta algo explixarlo por que al 100 no lo he entendido.



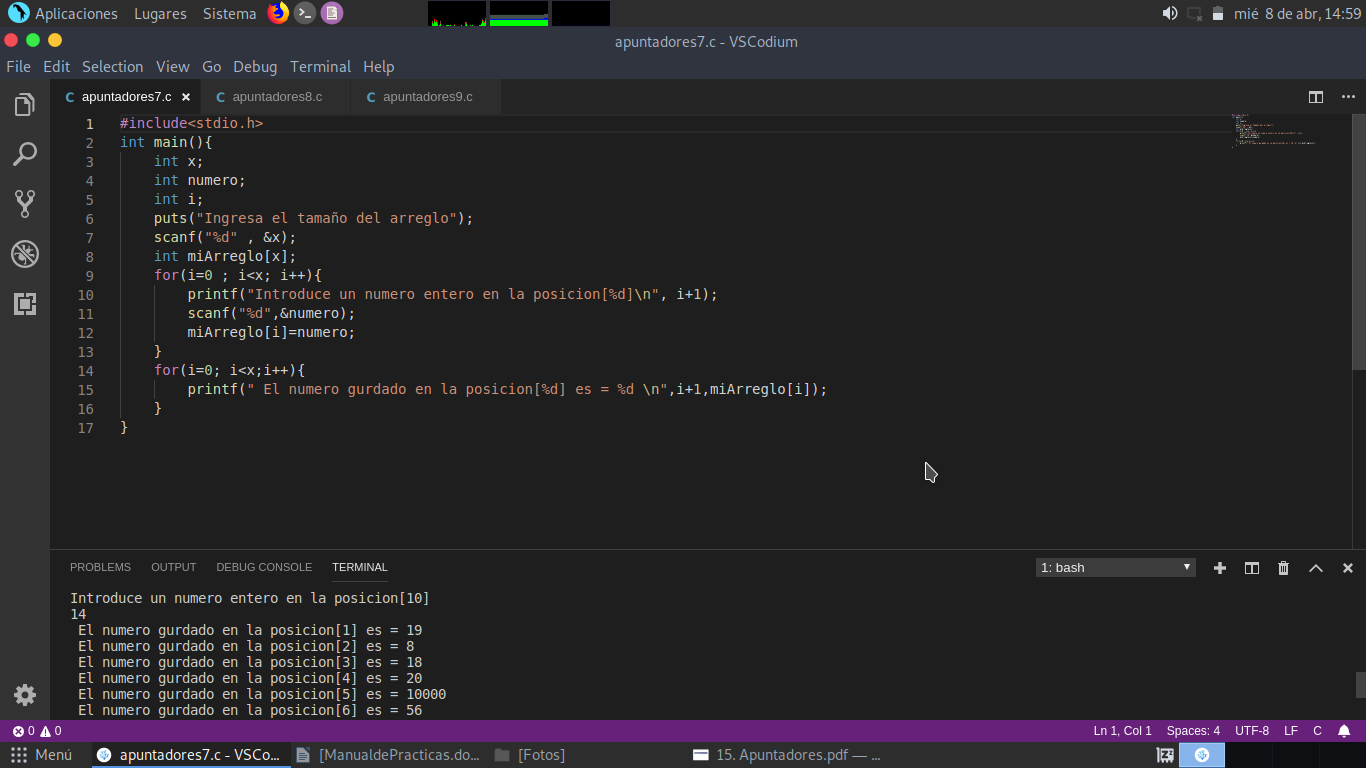
En la parte de abajo se muestra lo que imprime cuando lo corremos.



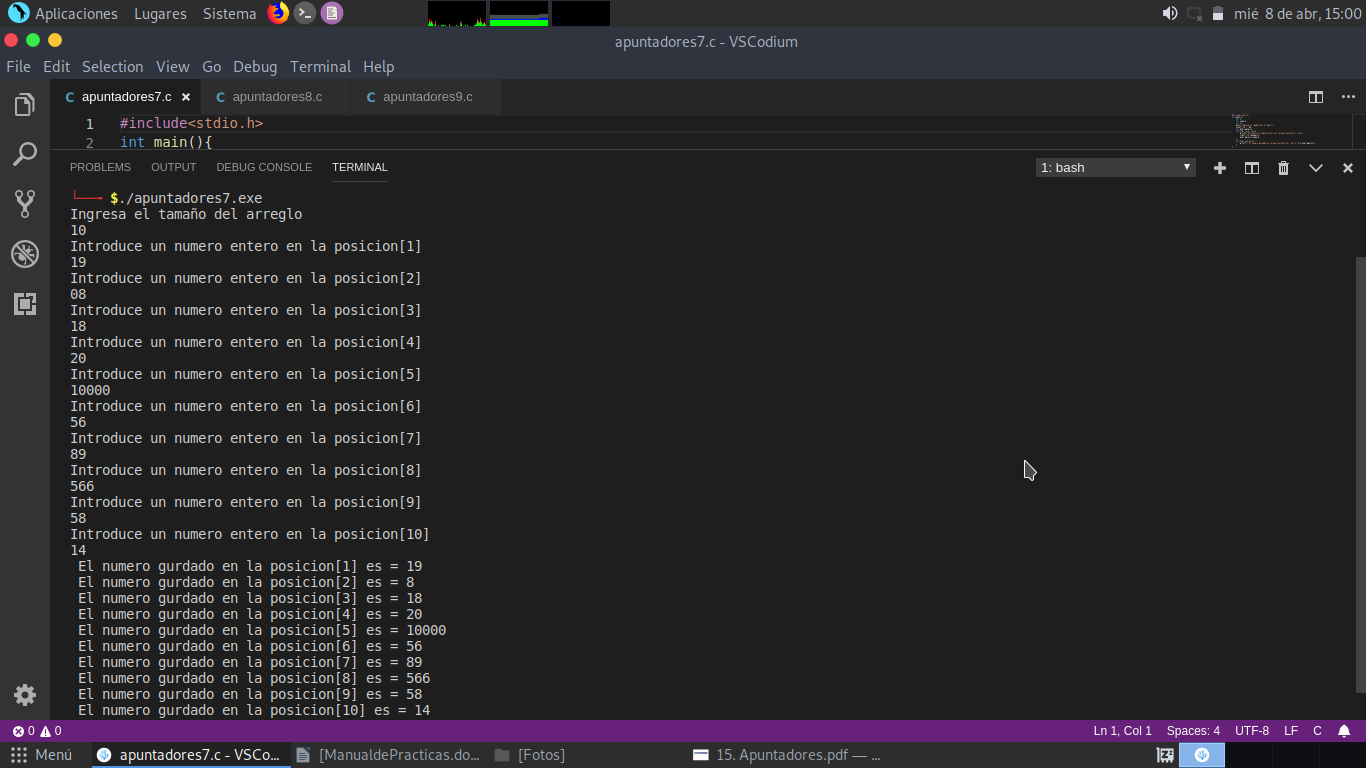
Ejercicio 7



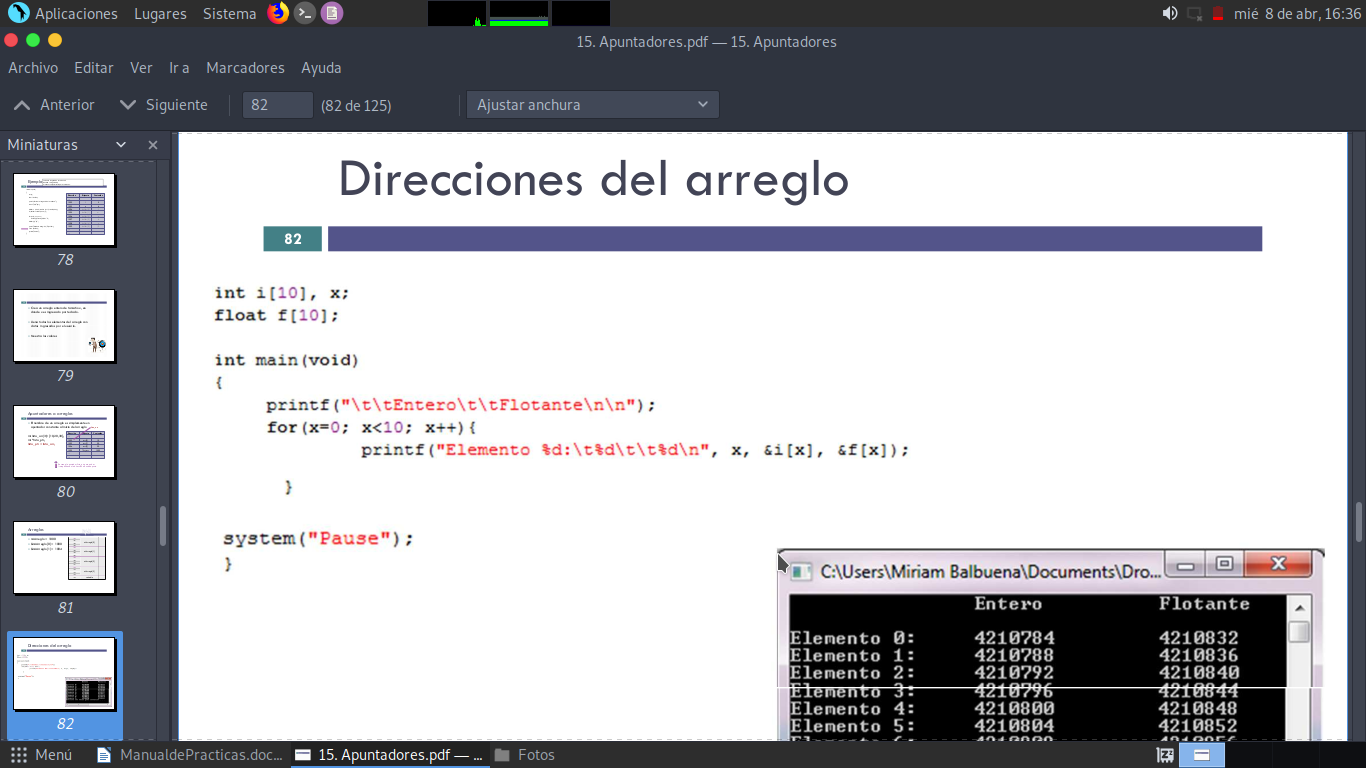
Arriba se muestra el ejercicio que vamos a resolver,que es crear un arreglo para eso declare una variable de tipo entero x otra de tipo entero numero y la ultima de tipo entero i, despues con ayuda de la funcion puts mando un mensaje para que le usuario ingrese el tamaño del arreglo, con ayuda del scanf pido que ingrese el tamaño por teclado y declaro mi arreglo. Para poder llenarlo lo hago con un for i lo inicializo en 0 la condicion es que i sea menos a i por que recordemos que se empieza de la posicion 0, dentro del for pedimos el numero y ese se guardara en el arreglo dependiendo de la posicion, con ayuda de otro for imprimo el arreglo.



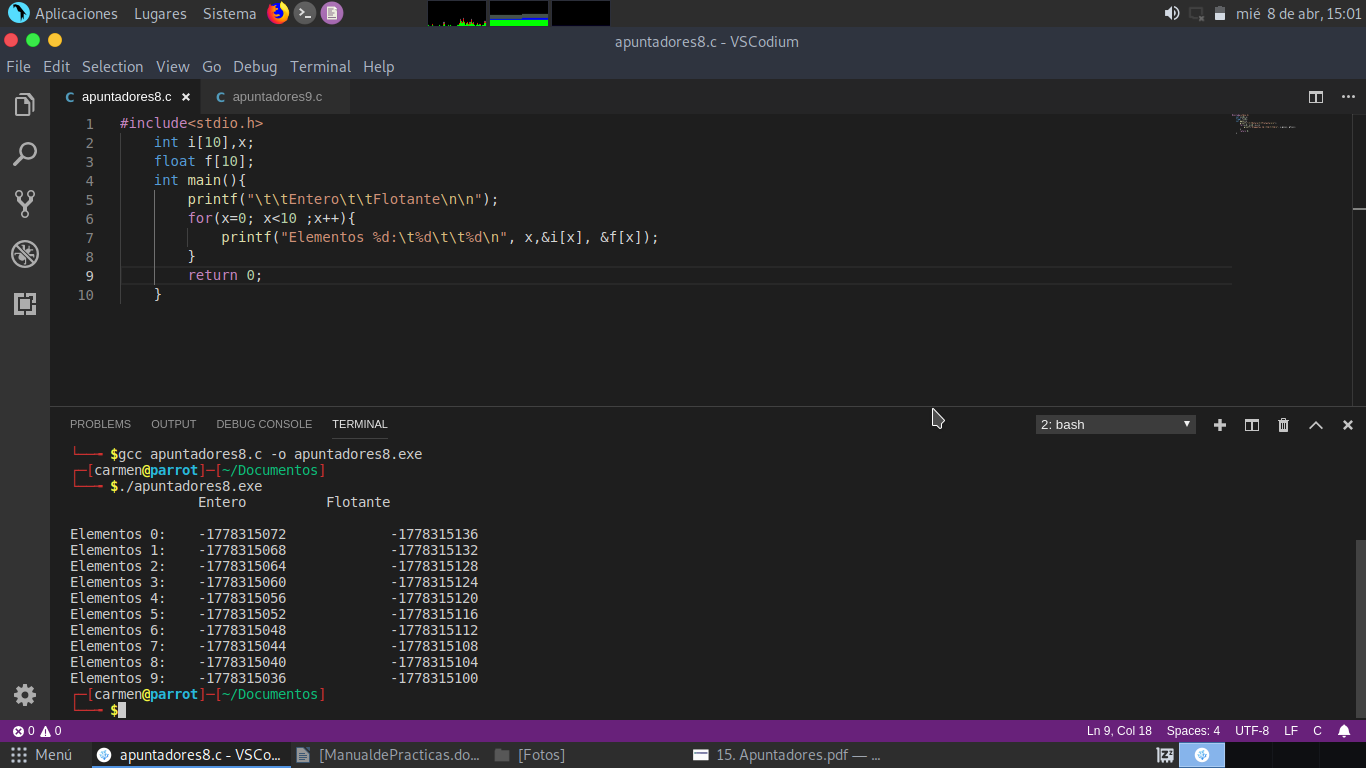
En la imagen de abajo se muestra se muestra cuando ingreso el tamaño del arreglo y se ve como pide los numeros e imprime lo que contiene en cada posicion.



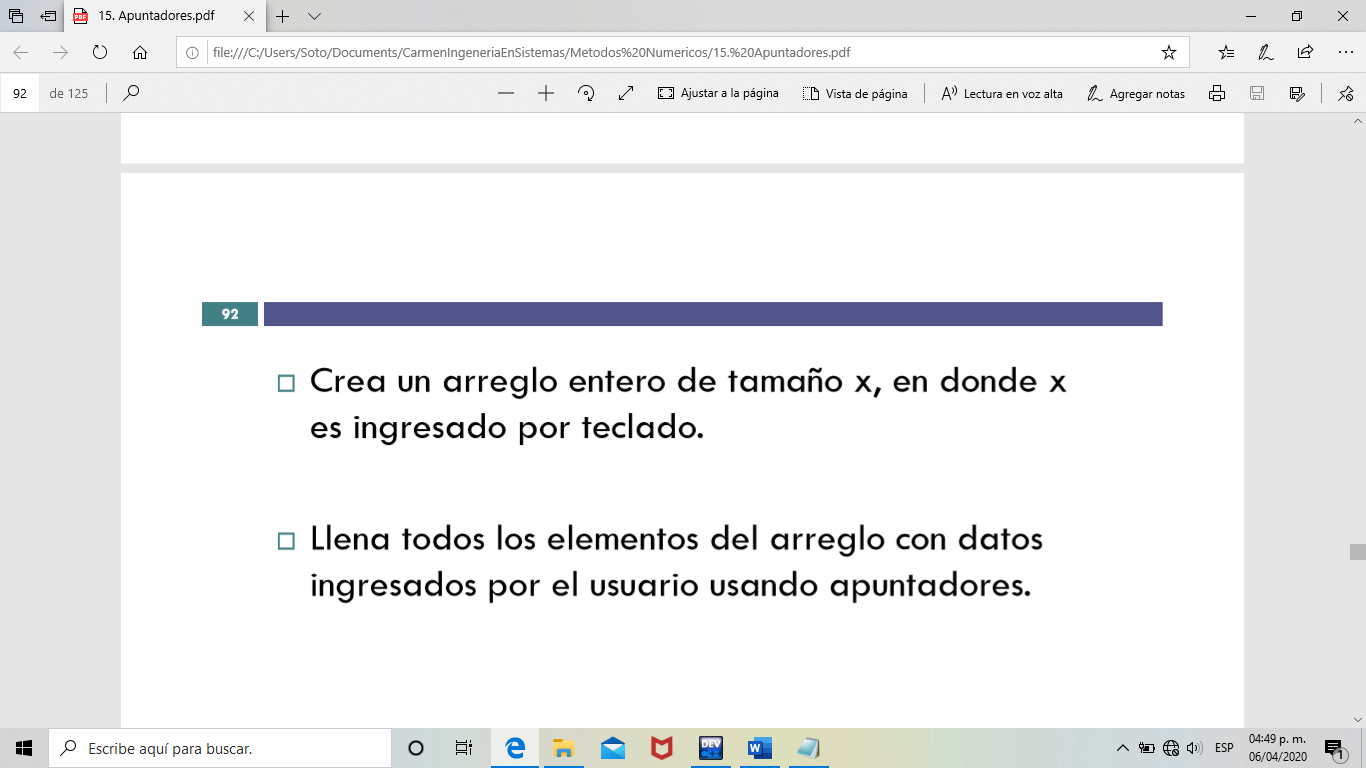
Ejercicio 8



Para verificar que esto fuera correcto hice el programa y lo ejecute par ver lo que nos diera, vi que fuera correcto y claramente si lo fue. Lo guarde y lo compile.

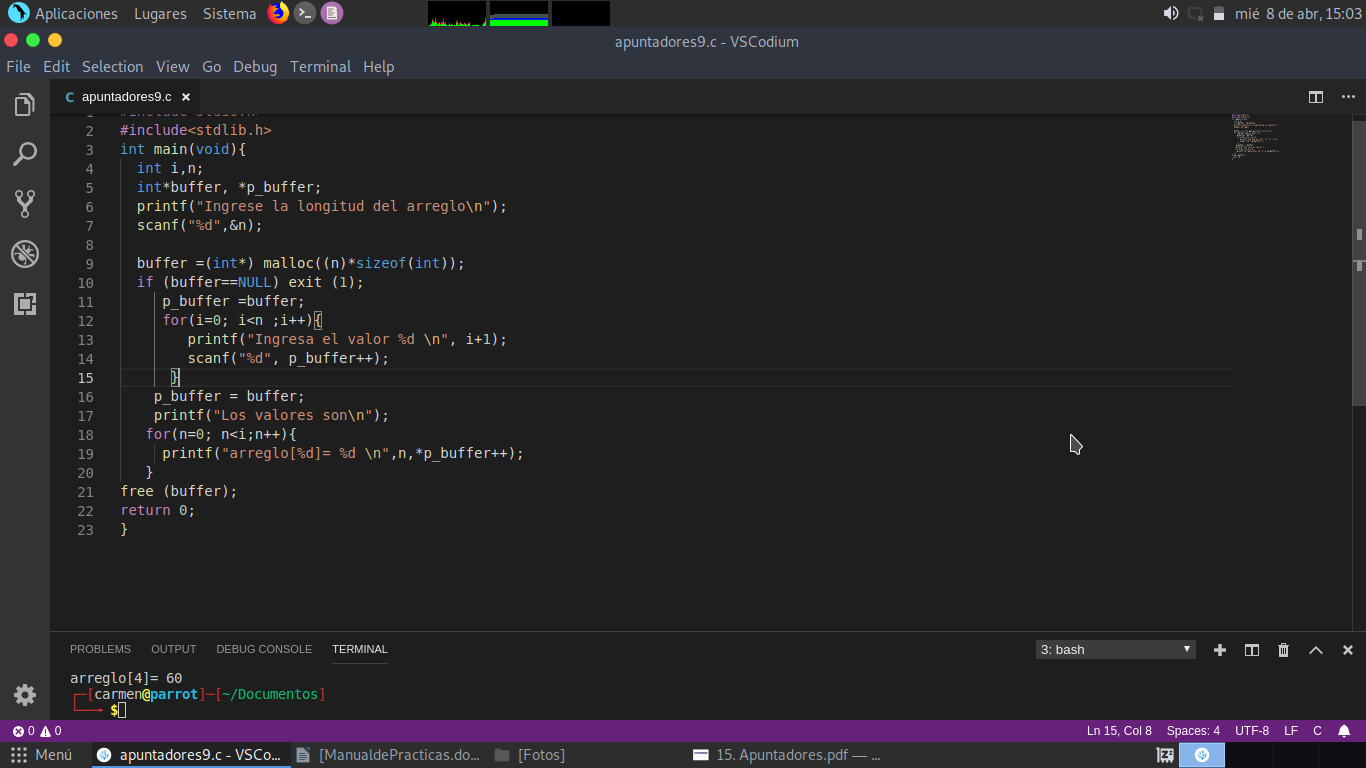


Ejercicio 9

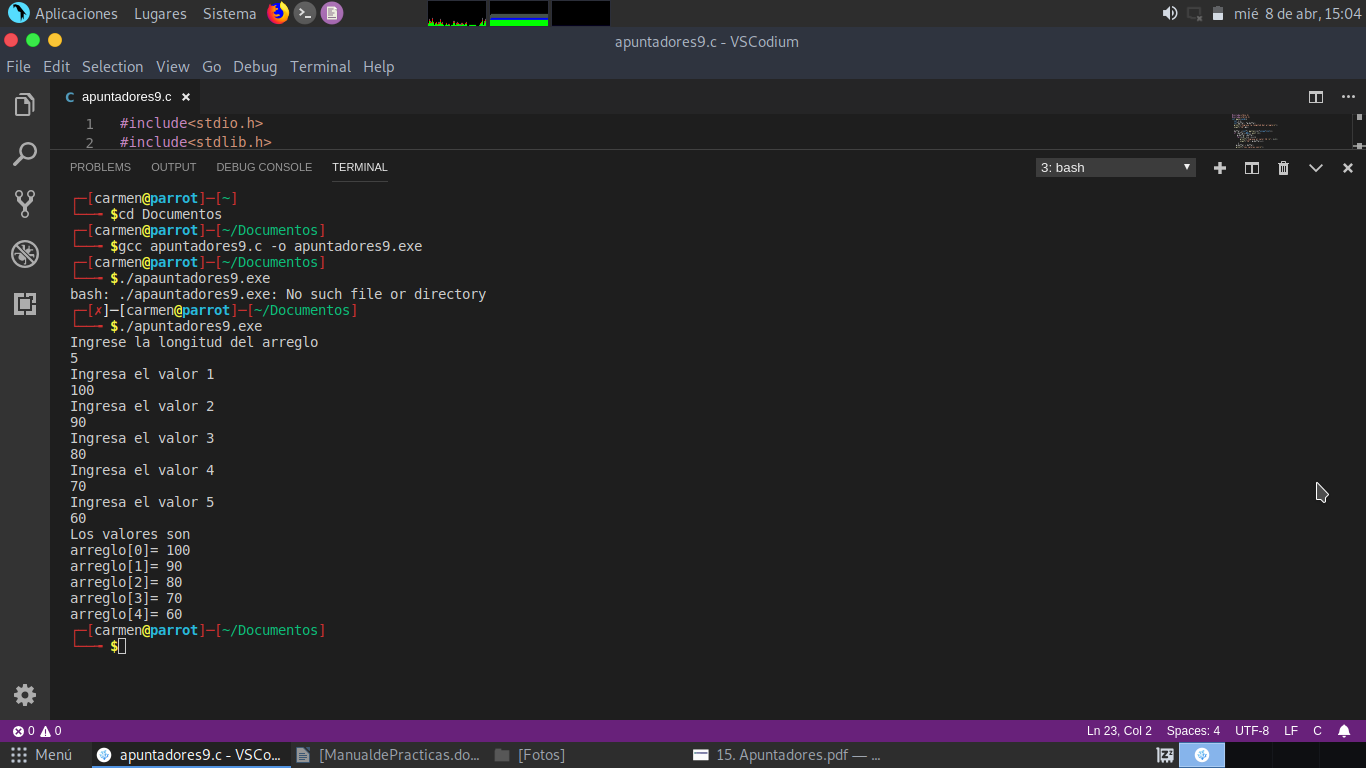


en la parte de abajo esta la solucion del ejercicio para que el usuario ingrese el tamaño del arreglo.

Con ayuda de un if comprobe el tamaño y lo que contenia, despues con ayuda de un for vamos a ingresar los datos y con otro ivamos a imprimir el arreglo con ayuda de apuntadores, en la imagen se muestra la estructura de este es mas practico mandar a imprimir los valores que ingrese el usuario.



En la imagen de abajo se muestra el tamaño del arreglo y los valores que ingresa el usuario asi mismo se muestra los valores guardados y estos se imprimirarn



**V. Conclusiones: En mi puento de vista son muy utiles los apuntadores por que nios ayudan acceder a las localidades de memoria tanto de datos primitivos como de arreglos. Bueno otra cosa es que son variables que guardan la direccion de otra variable. Los primeros ejercicios estubieron faciles pero ya a los ultimos sise me complicaron y mucho ya que me revolvi bastante.**